複単語表現検出における LLM ファインチューニングの有効性

井手 佑翼 [1] Joshua Tanner[2] Adam Nohejl[1] Justin Vasselli[1] 上垣外 英剛 [1] 渡辺 太郎 [1]
[1] 奈良先端科学技術大学院大学 [2]Resolve Research
ide.yusuke.ja6@is.naist.jp

概要

複単語表現、すなわちイディオム性を持つ単語の系列の検出は、機械翻訳など様々な下流タスクで重要な役割を果たす。本研究では、これまで複単語表現検出で用いられてこなかった大規模言語モデルのファインチューニングについて検証する。比較対象として、先行研究で優れた性能を示したエンコーダベースのシステムの検証も行う。実験の結果、提案手法が先行手法を大きく上回る性能を見せた。一方、提案手法を含む全手法で、再現率の低さが課題であることも明らかになった。また、モデルサイズを考慮した比較の結果から、メモリ効率の観点ではエンコーダベースの手法に利があることを示唆する。

1 はじめに

言語を理解するためには語彙が必要である。一般的に語彙の大部分は単語で構成されるが、複単語表現（multiword expression、MWE）も重要な一部をなす。MWE は、イディオム性を持つ単語の系列と定義される [1]。たとえば、次の文の太字部分が MWE に該当する[1)]。

(a) ***Pick** me **up** at the station.*

(b) *He has been **under the weather** lately.*

テキスト中の MWE を自動的にタグ付けする処理は、複単語表現検出（MWE identification、MWEI）と呼ばれる [2]。MWEI は、機械翻訳や語彙難解度推定など様々な下流タスクにおいて、重要な役割を果たすことがある [3, 4]。

近年、MWEI の関連タスクである固有表現抽出を含む自然言語処理タスクにおいて、大規模言語モデル（LLM）のファインチューニングを用いた手法が高い性能を示すと報告されてきた [5]。しかし MWEI では、管見のかぎり LLM の性能は検証されていない。これを受けて、本研究では、LLM ファインチューニングが MWEI にも有効かどうか検証する。実験で用いる LLM は、Llama [6] および Qwen [7] のファミリーから選ぶ。また、比較対象として、エンコーダベースの手法である MWEasWSD（MaW）[8] も検証する。MaW は、DiMSUM データセット [9] で最高精度を達成したモデルである。

実験の結果、ファインチューニングされた 72B パラメータの Qwen モデルが、MWEasWSD の各システムを大きく上回る性能を示した。一方で、すべてのモデル・システムが低い再現率（recall）に苦しむという課題もあることが明らかになった（例：Qwen-72B でも 50.7%）。エラー分析の結果、*real estate* など、WordNet [10] に含まれない MWE は、特に検出が難しいことが分かった。この事象の原因としては、これらの MWE は、MWE またはイディオムとして広く認識されていないことが考えられる。また、モデルサイズを考慮した比較では、最も性能のよい MaW システムは、数倍のメモリを要する Qwen-8B 等と同等の性能を実現することを示し、MaW はメモリ効率に優れることを示唆する。

2 手法

本研究で第一に検証する手法は、LLM ファインチューニングである。固有表現抽出 [5] や文法誤り訂正 [11] の先行研究で、LLM にタスクについての詳細な指示を訓練時と推論時に与える手法が、高い性能を示してきた。これらの研究と同様に、我々も、MWEI についての指示を LLM に与える手法を検証する。

第二の手法は、MaW [8] である。MaW は、MWE 辞書（WordNet [10]）とルールベースのパイプラインにより MWE の候補を列挙したうえで、訓練可能なエンコーダモデルにより候補をフィルタリングするシステムである [8]。著者が公開しているシステムのうち、我々は、双エンコーダ（bi-encoder）モデルを用いる Rule+BiEnc、および多エンコーダ（DCA

1) (a) の MWE は「〜を（車などで）迎えに行く」、(b) の MWE は「体の具合が悪い」を意味する。

表 1 CoAM の統計。

	文	MWE	MWE 内トークン比率 (%)
訓練	780	489	6.7
テスト	521	385	6.5
計	1,301	874	6.6

poly-encoder）モデルを用いる Rule+DCA を検証する。これらのモデルは、いずれも BERT [12] に基づくものである。エンコーダによるフィルタリングを省略したルールベースパイプライン（Rule）も公開されているため、これも検証する。

3 実験設定

3.1 モデル

LLM ファインチューニングには、Hugging Face で公開されている 4 つのインストラクションチューニング済みモデル、すなわち `Llama-3.1-8B-Instruct`、`Llama-3.1-70B-Instruct` [6]、`Qwen-2.5-7B-Instruct`、`Qwen-2.5-72B-Instruct` [7] を用いる。本論文では、バージョン情報を省略し、これらを Llama-8B のように表記する。また、訓練および推論を効率化するため、QLoRA [13] を用いる。ハイパーパラメータは、付録 A.1 に示すとおりである。

3.2 データセット

各モデル・システムの訓練および評価には、MWEI データセットの CoAM [14] を用いる。CoAM の統計は、表 1 に示すとおりである。CoAM の評価データに含まれる各 MWE には、その主辞の品詞に基づく MWE タイプの情報が付与されている。MWE タイプは、名詞、動詞、修飾語[2]・接続詞（修・接）、節、そしてその他の 5 種類である。4 節で、この MWE タイプを用いた分析を行う。なお、CoAM は、MWE を、意味的、文法的、または語彙的なイディオム性を持つ単語の系列と定義し、透明な（構成素から全体の意味を推測できる）コロケーションを MWE の定義から除外している。本研究における MWE の定義も、これに従うこととする。

3.3 定式化

CoAM は、MWEI をトークン単位の系列タグ付けとして定式化している [14]。すなわち、MWEI は、各文 s について、s に含まれるトークンの系列を入力として、その文に含まれる MWE のリストを出力するタスクである。ここで、各 MWE は、その MWE に含まれるトークンのインデックスのリストとして表される（表 7 参照）。なお、CoAM におけるトークンは、SpaCy によるトークナイズの結果として得られた単位である。

2) 形容詞または副詞。

表 2 `tsv_to_tsv` 形式に基づくプロンプトの例。`[...]` は、スペースの都合で省略した箇所である。

ロール	メッセージ
System	`You are a helpful system to identify multiple-word expressions (MWEs).`
User	`Identify all the MWEs in the given sentence, and output their surface forms and the indices of their components.\n` `\n` `Here, an MWE is defined as a sequence that satisfies the following three conditions.\n` `1. It consists of multiple words that are always realized by the same lexemes. [...]\n` `2. It displays semantic, lexical, or syntactic idiomaticity. [...]\n` `3. It is not a multi-word named entity, i.e., a specific name of a person, facility, etc.\n` `\n` `Each sentence is given as a string of words delimited by '\n'. Respond in TSV format, where the first and second columns contain words and MWE tags, respectively. The MWE tag should be a string of MWE identifiers. When a word belongs to multiple MWEs, the tag should be a concatenation of their numbers delimited by semicolons.\n` `\n` `Sentence:\n` `ACL\n` `stands\n` ⋮

3.4 評価指標

評価には、MWE ベース [15] の適合率、再現率、F1 スコアを用いる（正解の MWE と予測された MWE のそれぞれに含まれるトークンが、完全一致しているかどうかに基づいて評価する）。

3.5 予備実験

性能のよい入出力フォーマットを探るため、3 つのフォーマットを比較する予備実験を行った。実験の詳細は、付録 A.2 を参照されたい。結果として、`tsv_to_tsv` と呼ぶ形式（表 2 参照）のみが期待通りの出力を実現することが明らかになったため、以降の実験ではこの形式を用いる。

4 結果・分析

全体スコア 実験結果を、表 3 に示す。表 3a の左 3 列から、ファインチューニング済みの Qwen-72B（FT Qwen-72B）が最高の F1 値を実現したことが読み取れる。FT Qwen-72B は、適合率および再現率でもすべての MaW システムを上回った。これは、MWEI に対する LLM ファインチューニングの有効性、とりわけ多数のパラメータを持つ LLM のファインチューニングの有効性を示す結果と言える。この要因としては、LLM は事前学習を通じて MWE に関する知識を獲得しており、MWEI のためのファインチューニングを行うことでその知識を引き出すことができる、という説明が考えられる。

一方で、ルールベースパイプラインを除いて、すべてのモデル・システムは、低再現率に苦しむ結果となった。最善のシステムである FT Qwen-72B でも、再現率は 50.7%に留まり、検出すべき MWE のほぼ半分は検出できていないことが示された。この原因を、次に分析する。

再現率の分析 表 3a の右 4 列から、モデル・システムによらず、節 MWE や名詞 MWE は、修・接 MWE や動詞 MWE より検出が難しい傾向があることが分かる。また、表 3b の左 2 列からは、unseen である MWE（訓練データに現れない MWE）は、seen である MWE より検出しづらいこと、中央 2 列からは、非連続な MWE（構成素のトークンが完全に連続していない MWE）は、連続な MWE より検出しづらいことが読み取れる。

表 3b の右 2 列は、WordNet 内の（WordNet に含まれる）MWE は、WordNet 外の MWE より検出が難しいことを示している。MaW のルールベースパイプラインは、WordNet に含まれない MWE を候補として検出できないため、MaW のシステムが WordNet 外 MWE を一切検出できないのは当然の結果である。一方で、ファインチューニング済み LLM も、WordNet 外 MWE の検出性能が低くなった点は注目に値する。これを説明する仮説としては、WordNet に含まれる MWE は、比較的広く MWE またはイディオムとして認識されており、そのような認識が LLM の訓練データにも反映された結果、これらの MWE の検出性能が高くなった、と考えられる。

続いて、定性分析を行う。表 4 に、ランダムシードによる 3 回の Qwen-72B のファインチューニングの結果、すべての回で正しく検出された MWE と見逃された MWE の例を示す。*fire up* に代表される MWE 内の動詞 MWE は比較的容易に検出でき、FT Qwen-72B の再現率は 79.0%であった。一方、非連続な MWE である *in . . . hands* は正しく検出されなかった。また、名詞 MWE の *real estate* も見逃された。*real estate* は WordNet に含まれない MWE であるため、これは、前述の仮説と整合的な結果と言える。このような MWE の検出性能を高めることは、今後の課題である。

アブレーション FT Qwen-72B が高い性能を示したことを踏まえ、zero-shot 学習（ZSL）および few-shot 学習（FSL）との比較により、ファインチューニングがどの程度その性能に貢献したかを調査する。ZSL では、FT と同じプロンプトをモデルに与える。FSL では、入力（文）と出力（MWE のリスト）のペアを 5 つ訓練データからランダムに抽出し、それらを `tsv_to_tsv` 形式に変換したのち、プロンプトに事例として含める。この抽出プロセスでは、モデルがタスクと入出力形式について学ぶために十分な事例を与えるため、少なくとも 2 つの文が MWE を含むようなサンプルを得られるまで、抽出を繰り返す。モデルは、Qwen-72B と Llama-70B を用いる。

表 5 に、実験の結果を示す。ファインチューニングの性能は ZSL や FSL を大きく上回り、ファインチューニングの有効性を示す結果となった。

モデルサイズを考慮した比較 最後に、モデルのサイズ、すなわちパラメータ数を考慮したモデル・システム間の比較を行う。言語モデルのパラメータ数は、必要とされる GPU や推論時間を大きく左右するため重要である。MaW の各エンコーダ（`bert-base-uncased`）のパラメータ数はおよそ 110M なので、Rule+BiEnc および Rule+DCA は、（モデルのパラメータのみで）全体で $2 \times 110\text{M} \times 4 = 880\text{M}$ バイトの VRAM を要する（2 はエンコーダ数、4 は 1 パラメータあたりのバイト数）。一方、4 ビット量子化を施した Qwen-72B は、およそ $72,000\text{M} \times 0.5 = 36,000\text{M}$ バイト、すなわち MaW の約 40 倍の VRAM を要する。

表 3　モデル・システムごとの平均スコア（ランダムシードを用いた 3 回の訓練の結果の平均）。括弧内の数値は、そのカテゴリに含まれる MWE の数。± は標準偏差、太字は最善のスコア、WN は WordNet を表す。

(a)

					MWE タイプごと再現率			
		F1	適合率	再現率	名 (118)	動 (154)	修・接 (88)	節 (6)
FT	Llama-8B	$29.4_{\pm 2.1}$	$\mathbf{82.6}_{\pm 1.2}$	$17.9_{\pm 1.6}$	$5.9_{\pm 0.8}$	$26.4_{\pm 1.5}$	$24.2_{\pm 3.5}$	$0.0_{\pm 0.0}$
	Llama-70B	$38.4_{\pm 4.8}$	$74.5_{\pm 2.6}$	$26.1_{\pm 4.6}$	$19.2_{\pm 4.2}$	$35.7_{\pm 5.7}$	$25.4_{\pm 5.6}$	$0.0_{\pm 0.0}$
	Qwen-7B	$45.2_{\pm 0.6}$	$63.2_{\pm 0.9}$	$35.2_{\pm 0.8}$	$26.0_{\pm 2.0}$	$47.2_{\pm 1.0}$	$31.8_{\pm 2.0}$	$0.0_{\pm 0.0}$
	Qwen-72B	$\mathbf{55.5}_{\pm 0.5}$	$61.5_{\pm 2.6}$	$\mathbf{50.7}_{\pm 2.5}$	$\mathbf{44.4}_{\pm 3.2}$	$\mathbf{60.6}_{\pm 1.0}$	$\mathbf{50.4}_{\pm 5.1}$	$\mathbf{22.2}_{\pm 9.6}$
MaW	Rule	32.7	28.3	38.7	37.3	40.9	47.7	0.0
	Rule+BiEnc	$41.6_{\pm 0.1}$	$48.6_{\pm 0.3}$	$36.5_{\pm 0.3}$	$32.2_{\pm 0.0}$	$41.1_{\pm 0.7}$	$44.3_{\pm 0.0}$	$0.0_{\pm 0.0}$
	Rule+DCA	$42.0_{\pm 0.1}$	$48.4_{\pm 0.6}$	$37.1_{\pm 0.3}$	$33.3_{\pm 0.5}$	$40.9_{\pm 0.0}$	$45.8_{\pm 0.7}$	$0.0_{\pm 0.0}$

(b)

		Seen/Unseen 別再現率		連続／非連続別再現率		WN 内外別再現率	
		Seen (138)	Unseen (247)	連続 (335)	非連続 (50)	WN 内 (163)	WN 外 (222)
FT	Llama-8B	$37.2_{\pm 3.4}$	$7.2_{\pm 0.6}$	$20.6_{\pm 1.9}$	$0.0_{\pm 0.0}$	$29.2_{\pm 1.9}$	$9.6_{\pm 1.7}$
	Llama-70B	$35.0_{\pm 7.3}$	$21.1_{\pm 3.2}$	$28.7_{\pm 5.0}$	$8.7_{\pm 2.3}$	$46.8_{\pm 6.5}$	$10.8_{\pm 3.2}$
	Qwen-7B	$44.4_{\pm 0.8}$	$30.0_{\pm 0.8}$	$40.1_{\pm 0.9}$	$2.0_{\pm 0.0}$	$50.1_{\pm 1.4}$	$24.2_{\pm 0.5}$
	Qwen-72B	$\mathbf{58.2}_{\pm 5.8}$	$\mathbf{46.6}_{\pm 0.7}$	$\mathbf{56.3}_{\pm 2.9}$	$13.3_{\pm 1.2}$	$63.6_{\pm 3.4}$	$\mathbf{41.3}_{\pm 1.9}$
MaW	Rule	47.8	33.6	42.1	**16.0**	**91.4**	0.0
	Rule+BiEnc	$44.2_{\pm 0.0}$	$32.1_{\pm 0.5}$	$39.8_{\pm 0.3}$	$14.0_{\pm 0.0}$	$86.1_{\pm 0.7}$	$0.0_{\pm 0.0}$
	Rule+DCA	$45.4_{\pm 0.8}$	$32.4_{\pm 0.0}$	$40.5_{\pm 0.3}$	$14.0_{\pm 0.0}$	$87.5_{\pm 0.7}$	$0.0_{\pm 0.0}$

表 4　真陽性（TP）および偽陰性（FN）、すなわち Qwen-72B に正しく検出された／見逃された MWE の例。

結果	MWE	文脈	属性
TP	*fire up*	*The allegations have **fired up** the opposition, . . .*	WordNet 内、動詞
TP	*at least*	*. . . since **at least** the 1950s.*	WordNet 内、修・接
FN	*in . . . hands*	*. . . concentration of power **in** his own **hands**.*	WordNet 内、修・接
FN	*real estate*	*. . . park their toxic **real estate** assets . . .*	WordNet 外、名詞
FN	*you know*	***You know**, it's very old . . .*	WordNet 外、節

表 5　平均 F1 スコア（3 回の試行の結果の平均）。FT のランダム性は訓練時に、FSL と ZSL のランダム性は事例抽出に際して生じる。± は、標準偏差を表す。

	Llama-70B	Qwen-72B
FT	$38.4_{\pm 4.8}$	$55.5_{\pm 0.5}$
FSL	$4.3_{\pm 0.9}$	$14.1_{\pm 0.3}$
ZSL	6.9	2.8

また、4 ビット量子化を施した Qwen-7B は、MaW の約 4 倍の VRAM を要するが、その性能は Rule+DCA と同程度に過ぎなかった。この結果は、MaW の Rule+DCA は LLM よりメモリ効率がよいことを示唆する。さらに、MaW のボトルネックは MWE 辞書のサイズであることが特定されており [8]、この点を改善することで、Qwen-72B に匹敵する性能を実現できるかもしれない。

5　おわりに

本研究では、2 つの MWE 検出手法の性能を評価した。すなわち、LLM のファインチューニングと、エンコーダベースのシステム MaW を評価した。実験の結果、ファインチューニング済み Qwen-72B は MaW を大きく上回る性能を達成し、LLM ファインチューニングの有効性を示した。一方、メモリ効率の点では、MaW に利があるという示唆も得られた。

今後の研究の方向としては、LLM が MWE に関する知識を一定程度保持していることを踏まえ、LLM を活用して大規模な MWE 辞書を作成することが考えられる。この辞書をエンコーダベースのシステムと組み合わせることで、メモリ効率を抑えながら高い検出性能を実現できる可能性がある。

謝辞

本研究は、JST 次世代研究者挑戦的研究プログラム JPMJSP2140 の支援を受けたものである。また、本研究を進めるにあたり、Jacob Hoffman 氏から MWE に関する有益なコメントをいただいた。

参考文献

[1] Timothy Baldwin and Su Nam Kim. Multiword expressions. In **Handbook of Natural Language Processing**, 2010.

[2] Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke Van Der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. Multiword expression processing: A survey. **Computational Linguistics**, Vol. 43, No. 4, pp. 837–892, 2017.

[3] Eleftheria Briakou, Jiaming Luo, Colin Cherry, and Markus Freitag. Translating step-by-step: Decomposing the translation process for improved translation quality of long-form texts. In Barry Haddow, Tom Kocmi, Philipp Koehn, and Christof Monz, editors, **Proceedings of the Ninth Conference on Machine Translation**, pp. 1301–1317, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[4] Ekaterina Kochmar, Sian Gooding, and Matthew Shardlow. Detecting multiword expression type helps lexical complexity assessment. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 4426–4435, Marseille, France, May 2020. European Language Resources Association.

[5] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. UniversalNER: Targeted distillation from large language models for open named entity recognition. In **The Twelfth International Conference on Learning Representations**, 2024.

[6] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models, 2024.

[7] Qwen Team. Qwen2.5: A party of foundation models, September 2024.

[8] Joshua Tanner and Jacob Hoffman. MWE as WSD: Solving multiword expression identification with word sense disambiguation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Findings of the Association for Computational Linguistics: EMNLP 2023**, pp. 181–193, Singapore, December 2023. Association for Computational Linguistics.

[9] Nathan Schneider, Dirk Hovy, Anders Johannsen, and Marine Carpuat. SemEval-2016 task 10: Detecting minimal semantic units and their meanings (DiMSUM). In Steven Bethard, Marine Carpuat, Daniel Cer, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, **Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)**, pp. 546–559, San Diego, California, June 2016. Association for Computational Linguistics.

[10] George A. Miller. Wordnet: A lexical database for english. **Communications of the ACM**, Vol. 38, No. 1, pp. 39–41, 1995.

[11] Masahiro Kaneko and Naoaki Okazaki. Reducing sequence length by predicting edit spans with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 10017–10029, Singapore, December 2023. Association for Computational Linguistics.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[13] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 10088–10115. Curran Associates, Inc., 2023.

[14] Yusuke Ide, Joshua Tanner, Adam Nohejl, Jacob Hoffman, Justin Vasselli, Hidetaka Kamigaito, and Taro Watanabe. CoAM: Corpus of all-type multiword expressions, 2024.

[15] Agata Savary, Cherifa Ben Khelil, Carlos Ramisch, Voula Giouli, Verginica Barbu Mititelu, Najet Hadj Mohamed, Cvetana Krstev, Chaya Liebeskind, Hongzhi Xu, Sara Stymne, Tunga Güngör, Thomas Pickard, Bruno Guillaume, Eduard Bejček, Archna Bhatia, Marie Candito, Polona Gantar, Uxoa Iñurrieta, Albert Gatt, Jolanta Kovalevskaite, Timm Lichte, Nikola Ljubešić, Johanna Monti, Carla Parra Escartín, Mehrnoush Shamsfard, Ivelina Stoyanova, Veronika Vincze, and Abigail Walsh. PARSEME corpus release 1.3. In Archna Bhatia, Kilian Evang, Marcos Garcia, Voula Giouli, Lifeng Han, and Shiva Taslimipoor, editors, **Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)**, pp. 24–35, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

A　実験設定

A.1　ハイパーパラメータ

表 6　ハイパーパラメータ。エポック数は [5]、その他のパラメータは [13] を参考に決定した。

LoRA	r	64
	α	16
	Dropout	0.05
	Target modules	All linear layers
訓練	Epoch	3
	Effective batch size	32
	Learning rate	2e-4
	Learning rate scheduler	constant
	Optimizer	paged_adamw_8bit ($\beta_2 = 0.999$)
	Max grad norm	0.3

表 6 に、実験で用いるハイパーパラメータを示す。なお、推論時は貪欲デコーディングを行う。

A.2　入出力の形式

表 7　入出力フォーマット一覧。太字は本実験に採用されたフォーマット。

名称	入出力例

dict_to_dict_list	`{1: 'ACL', 2: 'stands',`
	`3: 'for', ...}`
	↓
	`[{'surface': 'stands for',`
	`'indices': [2, 3]}]`

str_to_str_number_span	`ACL stands for Association`
	`for Computational Linguists .`
	↓
	`ACL <1>stands for</1> Association`
	`for Computational Linguists .`

tsv_to_tsv	`ACL\n`
	`stands\n`
	`for\n`
	⋮
	↓
	`ACL\t\n`
	`stands\t1\n`
	`for\t1\n`
	⋮

性能のよい入出力フォーマットを探るため、表 7 に示す 3 つのフォーマットを用いて予備実験を行った。これらのフォーマットはすべて、不連続な MWE や、他の MWE と重なる MWE を含む、あらゆる形の MWE を表すことができる。実験では、これらのフォーマットおよび CoAM を用いて、Llama-8B と Qwen-7B を訓練・評価した。これらのフォーマットに応じて、プロンプトを変化させる。また、本実験と同じく、表 6 に示したハイパーパラメータを用いた。

実験の結果、`tsv_to_tsv` のみが、両モデルが従うことのできるフォーマットであることが明らかになった。その他のフォーマットについては、モデルがフォーマットに違反する結果を出力する結果となった。`dict_to_dict_list` を用いると、モデルは頻繁に誤った（トークンの）インデックスを出力した。`str_to_str_number_span` を用いると、モデルは句読点の前のスペースを消去した。各フォーマットは、フォーマットに合わせて調整されたプロンプトとともに用いられたことを踏まえれば、これは筆者の予想に反する結果であった。